

## ESTRUCTURA DE DATOS PROYECTO

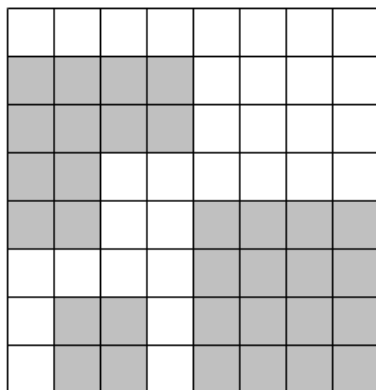
**Profesores:** Carlos Contreras Bolton – Felipe Reyes González

**Fecha:** 24 de Mayo de 2014

**Ayudantes:** Daniela Ubilla – Tamara Sáez

### Enunciado

Se pretende crear una nueva estructura, la que llamaremos NE. La cual representará una imagen digital en blanco y negro, donde una imagen puede ser representada en un espacio rectangular compuesto de  $N \times N$  píxeles (en potencia de a dos), cada uno representando un punto (blanco o negro) de la imagen. La representación usual de una imagen es una matriz de  $N \times N$ , con ceros los píxeles blancos y con 1 los negros, como se ve en la Figura 1.



0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0
1	1	0	0	0	0	0	0
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	1	1	0	1	1	1	1
0	1	1	0	1	1	1	1

Figura 1: Representación de una imagen en matriz

NE es un árbol 4-ario que permite representar de manera compacta una imagen, que en otro caso podría ocupar grandes cantidades de memoria. NE tiene tres tipos de elementos: blancos, negros y grises, que representan un grupo de píxeles de la imagen según el esquema recursivo de representación planteado en la Figura 2, en a) blancos, b) negros y c) grises.

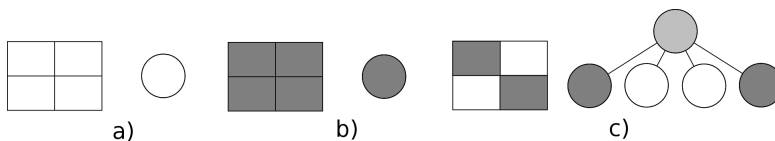


Figura 2: Esquema recursivo de representación

La raíz del NE representa la imagen completa: si la imagen es toda negra, el NE tiene un solo elemento con valor NEGRO. Si la imagen es completamente blanca, tiene un solo elemento con valor BLANCO.

En cualquier otro caso, la imagen se representa con un nodo GRIS, y se le asocian 4 subárboles, cada uno representando uno de los cuadrantes resultantes al dividir la imagen en 4 cuadros iguales. Al continuar este proceso recursivamente se obtiene la representación completa de la imagen por medio de una estructura arborescente compacta.

A modo de ejemplo, para la imagen que aparece en la Figura 1, la raíz del NE que la representa debe ser un nodo gris, para indicar que no todos los píxeles de la imagen son del mismo color. Luego, se divide la imagen en 4 cuadrantes y se aplica recursivamente el mismo proceso, asociando NE resultantes como subárboles, como se aprecia en la Figura 3.

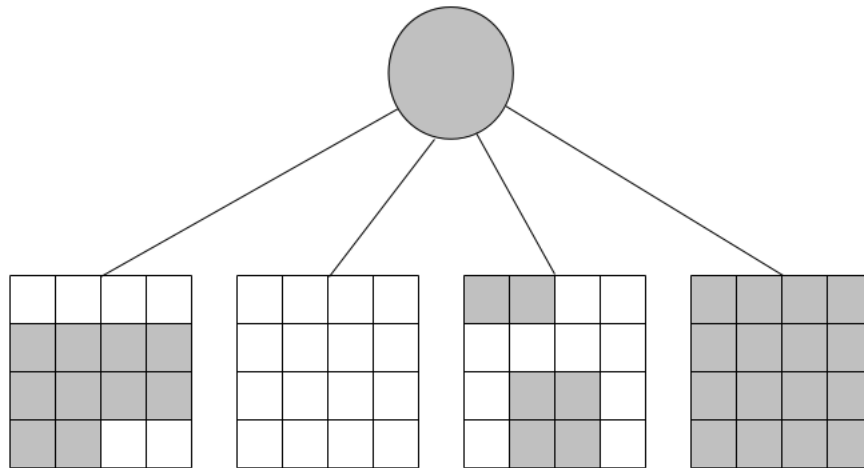


Figura 3: Esquema recursivo de representación

Este proceso sigue así, hasta llegar al siguiente árbol 4-ario, que representa la imagen completa, el cual se visualiza en la Figura 4.

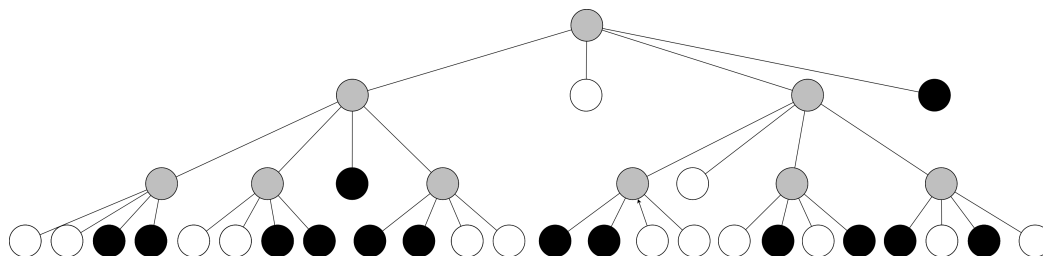


Figura 4: Conversión de la matriz a la estructura NE

Se solicita crear un programa, en el lenguaje C, que permita visualizar la imagen representada en el archivo `datos.dat`. Considere una matriz de  $100 \times 100$ . El archivo tiene códigos 0 (BLANCO), 1 (NEGRO), y 2 (GRIS), y el recorrido es por niveles.

Además, se solicita la implementación de las siguientes operaciones del NE sobre la estructura de datos sugerida son:

- La operación analizadora `valorPixel` considera dos casos: si la raíz tiene un color diferente a GRIS (BLANCO o NEGRO), el píxel tiene dicho valor. En caso contrario, localiza el cuadrante que contiene el píxel y desciende recursivamente.
- La función `cuadrante` es una rutina de utilidad, que retorna el cuadrante en el cual el píxel `[fil,col]` dentro del NE. Para esto divide en dos el rango de valores en la coordenada  $x$  y el rango de valores en la coordenada  $y$ , y determina en cuál de los cuatro cuadrantes está el píxel buscado.
- Una rutina que coloca un píxel en blanco, considera tres casos: si la raíz es GRIS, localiza el cuadrante en el que se encuentra el píxel y hace una llamada recursiva sobre dicho subárbol. Si la dimensión del NE es  $1 \times 1$  (solo tiene un píxel), lo coloca en BLANCO. Si la raíz del NE es de color

NEGRO, la cambia por gris, crea los 4 subárboles correspondientes de color NEGRO, y finalmente hace una llamada recursiva sobre el subárbol que contiene el píxel. La rutina crear se encarga de construir un nodo de color NEGRO con las coordenadas dadas en los parámetros.

- Una función que compare dos regiones de una imagen y encuentre qué hay de común entre ellas. Esto implica la intersección de dos imágenes y determinar los puntos comunes entre estas.

## 1. Salida

Su programa deberá generar dos tipos de salidas:

### 1.1. Por pantalla

En pantalla, si lo que se pide es un dato o un grupo de píxeles u otra información se muestran en conformidad a los píxeles con números 0 para BLANCO y 1 para NEGRO y 2 para GRIS. Cuando tenga que mostrar un árbol, se usará el mismo formato de entrada.

### 1.2. Archivo

Cuando se realicen modificaciones a los árboles deberá generar un archivo que represente el árbol. Utilice el lenguaje dot (Graphviz permite mostrar en dibujo <http://www.graphviz.org/>).

## 2. Condiciones del proyecto

- El Proyecto es en grupos de máximo dos personas.
- Dudas respecto al trabajo, realizar a los ayudantes de manera personal o vía correo electrónico.
- El lenguaje a utilizar es C de manera estándar. No se aceptarán bibliotecas no estándares. Su programa va a ser recompilado para su evaluación.
- Cree las bibliotecas o utilice las bibliotecas dadas por el profesor o ayudantes.
- En el manual de usuario, asuma que el usuario sabe instalar programas, sólo diríjalo a la página o indíquele cómo conseguir el (los) programa(s) necesario(s).
- **DEBE** ir archivo Makefile. Se utilizará para compilar. Si no compila con Makefile tendrá un 1 por compilación.
- Se deben entregar documentos complementarios al programa.
- El informe que debe entregar, tiene que mostrar claramente las razones de las decisiones que usted tomo para implementar la estructura NE, tanto de navegación, interfaz de usuario, funcionalidades que usted presenta. Además, el informe debe mostrar las decisiones tomadas con respecto a la implementación y uso del lenguaje.
- El manual de uso debe ser totalmente fiel a la versión entregada del programa, el cual será revisado siguiendo dicho manual, esto significa que si algo no está explicado en el manual de uso, no será revisado en el programa, cabe mencionar que de ninguna manera significa que no se intentará interactuar con el programa de una manera diferente a lo mencionado en el manual, por ejemplo, si el manual indica ingresar sólo números, no significa que no se intentará ingresar letras.
- Los documentos deben ser entregados en formato digital (pdf, no se admite otro formato) e impreso en papel el mismo día del laboratorio siguiente a la entrega del proyecto.
- El sistema debe ser robusto, se penalizarán las caídas de cualquier tipo.

- Se recomienda comprimir el directorio del proyecto. Se recompilará para su funcionamiento. Utilice bibliotecas estándares (La función `system` es estándar, pero realiza llamadas al sistema que no son estándares, así que no la utilice), si se encuentra una biblioteca no estándar, se comentará y se recompilará el programa.
- El archivo debe ser enviado al correo electrónico del ayudante, donde el asunto debe decir: Proyecto de ED 2014-01. Debe estar adjunto un archivo comprimido que contendrá el directorio del proyecto y los informes. El archivo y el nombre de la carpeta debe tener el siguiente formato:
  - ApellidoPaternoAlumno1ApellidoMaternoAlumno1\_ApellidoPaternoAlumno2ApellidoMaternoAlumno2.extensión.  
Ejemplo:  
Alan Brito Delgado y Armando Casas Segura: BritoDelgado\_CasasSegura.tar.gz.
  - En caso de no cumplir estos puntos será sancionado con 1,0 punto menos.
- Fecha de entrega **24 de Junio** hasta las 23:59 hrs.
- En la última semana del semestre será fijado un control de salida. Donde la nota corresponde al 10 % en caso de ser mayor o igual a un 4,0. En caso contrario, corresponde al 100 %.
- El enunciado podrá sufrir modificaciones que serán debidamente informadas.
- En caso de trabajar directamente con una imagen real, osea hacer la conversión de la imagen a la matriz tendrá bonificación extra.